

CONSTITUTION: Plural program counters 4 where addresses of plural independent programs are held and schedulers to which data indicating the selection order of program counters 4 is set are provided. A setting part 1 which sets the selection order of program counters 4 to schedulers and a selecting part 5 which repeatedly selects program counters 4 in accordance with the operation order set to schedulers are provided. Thus, the program meeting a need is quickly executed, and the processing time is distributed in accordance with the program, and the efficiency of the whole of the system is improved. Further, the processing time of the program is clarified, and real time (parallel) processing in a microprocessor is easily realized.

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平5-46413

(43)公開日 平成5年(1993)2月26日

(51)Int.Cl.⁵

G 0 6 F 9/46

識別記号

3 4 0 E 8120-5B

3 1 3 E 8120-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数3(全 5 頁)

(21)出願番号

特願平3-200517

(22)出願日

平成3年(1991)8月9日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(71)出願人 000221199

東芝マイクロエレクトロニクス株式会社

神奈川県川崎市川崎区駅前本町25番地1

(72)発明者 瀬頭 公四

神奈川県川崎市川崎区駅前本町25番地1

東芝マイクロエレクトロニクス株式会社内

(72)発明者 植村 幸政

神奈川県川崎市川崎区駅前本町25番地1

東芝マイクロエレクトロニクス株式会社内

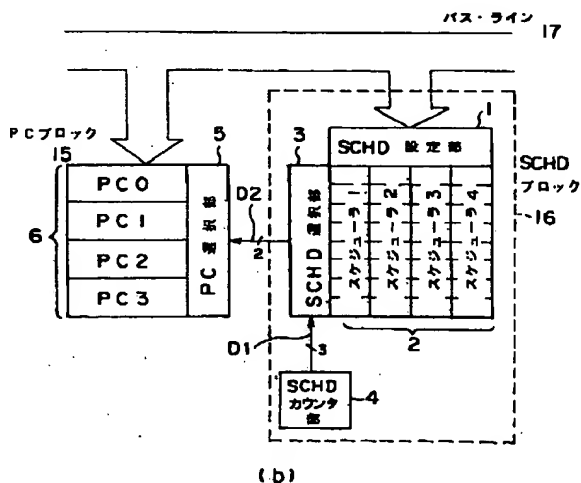
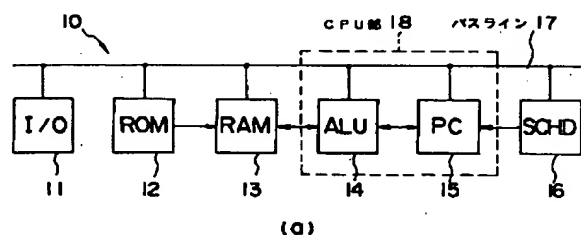
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 マイクロプロセッサ

(57)【要約】

【目的】 本発明は、時間の制限を受ける処理を含むプログラムの作成を容易に出来、かつシステムの向上が図れるマイクロプロセッサを提供することである。

【構成】 マイクロプロセッサは、独立した複数のプログラムのアドレスを保持する複数のプログラムカウンタと、プログラムカウンタの選択順序を指示するデータが設定されるスケジューラと、スケジューラにプログラムカウンタの選択順序を設定する設定部と、プログラムカウンタをスケジューラに設定された動作順序に従って、繰り返し選択する選択部を具備することを特徴とする。



【特許請求の範囲】

【請求項1】 独立した複数のプログラムのアドレスを保持する複数のプログラムカウンタ手段と、前記プログラムカウンタ手段の選択順序を指示するデータが設定されるスケジューラと、前記スケジューラに前記プログラムカウンタ手段の選択順序を設定する設定手段と、前記プログラムカウンタ手段を前記スケジューラに設定された動作順序に従って、繰り返し選択する選択手段を具備することを特徴とするマイクロプロセッサ。

【請求項2】 前記スケジューラが複数備えられており、前記選択手段は1つのスケジューラを選択することを特徴とする請求項1記載のマイクロプロセッサ。

【請求項3】 前記選択手段は、クロック信号を出力するカウンタ部と、前記カウンタ部からのクロック信号を受けて前記スケジューラに設定されたデータを選択するスケジューラ選択部と、前記スケジューラ選択部からの出力信号を受けるプログラムカウンタ選択手段から構成されることを特徴とする請求項1記載のマイクロプロセッサ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、互いに他から独立した2つまたはそれ以上のプログラムの同時実行に適したマイクロプロセッサのシステム構成に関するものである。

【0002】

【従来の技術】 従来のマイクロプロセッサとして、割り込み方式、タイムスライス方式、マルチプロセッサ方式等がある。

【0003】 割り込み方式とは、計算機のプログラム実行中に特定の割り込み信号が発生された場合、実行中のプログラムを中断し、その事象を処理する別のルーチンへ分岐して、その実行終了後、再び元のルーチンへ戻って実行を継続させる方式をいう。

【0004】 タイムスライス方式とは、複数個のプログラムを設定された優先順位で一定時間（スケジューラがプログラムに割り当てる中央処理装置の使用時間）おきに実行する方式をいう。

【0005】 マルチプロセッサ方式とは、複数のCPUを設置して、並列に演算を行うことで、高速性、信頼性、拡張性の点で優れた高性能の計算機システムを実現する方式である。

【0006】

【発明が解決しようとする課題】 従来の割り込み方式、タイムスライス方式、マルチプロセッサ方式等によって構成されるマイクロプロセッサにおいては、以下のような問題があった。

【0007】 まず、割り込み処理方式のマイクロプロセッサでは、特定のプログラムを処理させる為に別のルーチンへ分岐している間、他のプログラムの実行を中断す

る必要があった。その為、タイムロスを生じ、マイクロプロセッサの実行速度を低下させるという問題があった。

【0008】 次に、タイムスライス方式のマイクロプロセッサでは、スケジューラがプログラムに割り当てる中央処理装置の使用時間を大まかに（ms単位）分割する為、時間的制約を受けるという問題があった。

【0009】 次に、マルチプロセッサ方式のマイクロプロセッサでは、並列演算処理用のCPUが複数個必要の為、コストが高くなると共にCPUの管理とCPUの効率が一般的に高くなるという問題があった。

【0010】 本発明は、上記実情に鑑みてなされたもので、時間の制限を受ける処理を含むプログラムの作成を容易に出来、かつシステムの向上が図れるマイクロプロセッサを提供することである。

【0011】

【課題を解決するための手段】 本発明のマイクロプロセッサは、独立した複数のプログラムのアドレスを保持する複数のプログラムカウンタと、前記プログラムカウンタの選択順序を指示するデータが設定されるスケジューラと、前記スケジューラに前記プログラムカウンタの選択順序を設定する設定部と、前記プログラムカウンタを前記スケジューラに設定された動作順序に従って、繰り返し選択する選択部を具備することを特徴とする。

【0012】

【作用】 上記構成により、前記設定部は複数の前記プログラムカウンタの選択順序を前記スケジューラに設定する。前記選択部は、前記スケジューラに設定されたプログラムカウンタの選択順序に従って複数のプログラムカウンタを繰り返し選択する。

【0013】

【実施例】 以下、図面を参照しながら本発明に係るマイクロプロセッサについて説明する。

【0014】 本実施例のマイクロプロセッサは、疑似的に4つのCPUを内蔵したものである。物理的にはCPUは1つであるが、これが時分割で交互にプログラムを実行する。よって、リアルタイム制御用として使用するのに便利である。まず、図1に示すマイクロプロセッサ10の構成について説明する。図1(a)は、本発明の実施例に係るマイクロプロセッサのブロック図である。図1(b)は、図1(a)に示すPCブロック15とスケジューラブロック16の詳細な状態を示すブロック図である。図1(a)のマイクロプロセッサ10は、1チップ内に備えられる。

【0015】 図1(a)に示すマイクロプロセッサ10は、I/O11、ROM12、RAM13、ALU14（Arithmetic Logical Unit）、PC（プログラムカウンタ）ブロック15とスケジューラ（以下、SCHED）ブロック16を有する。I/O11、ROM12、RAM13、ALU14、PC

3

ブロック15とSCHDブロック16は、バスライン17に接続されている。

【0016】また、ROM12とRAM13、RAM13とALU14、ALU14とPCブロック15、PCブロック15とSCHDブロック16はそれぞれ直接接続されている。I/O11、ROM12、RAM13は従来のI/O、ROM、RAMと同様の機能を有する。ALU14とPCブロック15はCPU部18を構成している。

【0017】図1(b)に示されるようにPCブロック15は、PC選択部5とPC群6を有する。PC群6は、PC0、PC1、PC2、PC3の4つのPCから構成される。各PC(PC0、PC1、PC2、PC3)は、次に実行する命令のアドレスを保持する為の専用レジスタである。PC選択部5は、PC(PC0、PC1、PC2、PC3)を選択する為のものである。

【0018】SCHDブロック16は、SCHD設定部1、SCHD選択部3、SCHDレジスタファイル部2とSCHDカウンタ部4を有する。SCHDブロック16は、SCHDのデータ設定(2ビット×8個×4ブロック)と、SCHDの指定をする為のものである。

【0019】SCHDレジスタファイル部2は、8個のレジスタから成る4つのSCHD(SCHD1、SCHD2、SCHD3、SCHD4)から構成され、PC群6の動作順序を設定する為のものである。

【0020】尚、SCHDレジスタファイル部2は、最大4つ(SCHD1乃至4)まで使用出来る。SCHD設定部1は、SCHDレジスタファイル部2の各SCHD(SCHD1乃至4)にPC(PC0乃至3)の動作順序のデータを設定する。

【0021】SCHD選択部3は2ビットのデータ信号をデータラインD2を介してPC選択部5に出力している。SCHD選択部3は、SCHDレジスタファイル部2の各SCHD(SCHD1乃至4)に設定されたデータを選択する。SCHDカウンタ部4は3ビットのデータ信号をデータラインD1を介して、SCHD選択部3に供給している。I/O11は、データの入出力を行う入出力インターフェイスである。次に、図面を参照しながら上記実施例のPCを動作させるまでのマイクロプロセッサ10の動作について説明する。マイクロプロセッサ10の動作について説明する前に、動作説明に使用する図について説明する。図2(a)は、マイクロプロセッサにおけるPCの実行順序の一例を示す図である。例えば、図2(a)のPC3はプログラムカウンタ3を示している。図2(b)は、図2(a)に示すPCの動作順序の一部を示す図である。

【0022】図2(c)は、SCHDカウンタ部4からSCHD選択部3に出力される信号を示す図である。例えば、2ビット信号「11」はPC3を示しており、2ビット信号「00」はPC0を示している。図2(d)

4

は、SCHDカウンタ部からSCHD選択部3に出力される信号を示す図である。例えば、0はSCHD1内の第1番目のレジスタを示している。

【0023】図2(e)は、PCの実行例を示すデータテーブルである。例えば、図2(e)の(1)乃至(4)は、SCHD1に設けられた8つのレジスタに設定されたデータを示している。例えば、図2(e)の(2)の3ビット信号「3」は第1番目のレジスタに設定されたPC3、3ビットの信号「2」は2つ目のレジスタに設定されたPC2を示している。次に、マイクロプロセッサ10の動作について説明する。まず、プログラム上でバスライン17を介してSCHD設定部1にデータが伝送される。

【0024】SCHD設定部1は、バスライン17からのデータを受けて、図2(e)に示すようにSCHD(SCHD1乃至4)にPCの(PC0乃至PC3)の実行順序を設定する。

【0025】具体例で説明すると、SCHD設定部1は、SCHD1には図2(e)の(1)に示すデータを設定し、SCHD2には、図2(e)の(2)に示すようなPC(PC0乃至3)の動作順序を設定し、SCHD3には、図2(e)の(3)に示すようなPC(PC0乃至3)の動作順序を設定し、また、SCHD4には図2(e)の(4)に示すようなPC(PC0乃至3)の動作順序を設定する。

【0026】一方、SCHD選択部3には図示せぬデータラインよりどのスケジューラを選択すべきかを指示するデータがプログラム等により供給される。SCHD選択部3は、指示に従って、例えばSCHD2を選択する。

【0027】SCHD設定部1でPC(PC0乃至3)の動作順序が設定されると、さらに、どのスケジューラを選択すべきかが設定されると、常に3ビットのセレクト信号D1がSCHDカウンタ部4からSCHD選択部3に伝送されているので、SCHD選択部3が選択されたSCHDに設定された8つのデータのいずれかを順番に選択する。例えば、SCHD2が選択された場合には、図2(e)の(2)、即ち、図2(b)に示される一連のデータが順番に選択される。SCHD選択部3は選択した図2(c)に示されるデータ2ビットのデータ信号をPC選択部5に出力する。

【0028】SCHD選択部3からのデータ信号を受けて、PC選択部5が図2(a)に示すようにPC3、PC2、PC3、PC1、PC3…の順にPC(PC0乃至3)を選択する。図2(a)に示すようにPC0まで選択されると、またPC3が選択される。上記動作より、図2(a)に示すような動作順序でPCに保持されたアドレスに従って、プログラムが実行される。尚、SCHD1が設定された場合には、プログラムカウンタPC0だけが実行され、他のPC(SCHD1乃至3)は

5

実行されない。図1のマイクロプロセッサ10は、バスライン17を介して動作終了の命令を受けると、その動作を終了する。次に、図面を参照しながら各PCに保持されたアドレスに従って実行されるプログラムの実行速度について説明する。例えば、SCHD2に図2(a)のようにPCの実行順序が設定されている場合には、各PCに従うプログラムの実行速度は以下になる。図2に示すように、一命令に割り当てられている中央処理装置の使用時間 T_2 を $1\mu s$ と仮定する。一命令の実行の中でPCは8回実行されるので、PCの1回の動作時間は $1/8\mu s$ で求められ、 T_1 は $0.125\mu s$ となる。

【0029】よって、図2(b)のようにPCの動作順序が設定されている場合には、1命令の中でPC0、PC1は1回実行されるので、PC0に割り当てられる中央処理装置の使用時間は $0.125\mu s$ であり、その使用速度は $1.000\mu s/INST$ である。

【0030】PC2は2回実行されるので、PC2における中央処理装置の使用時間は $0.25\mu s$ であり、その使用速度は $1/2$ で求められ、 $0.500\mu s/INST$ である。

【0031】PC3は4回実行されるので、PC3における中央処理装置の使用時間は $0.5\mu s$ であり、その使用速度は $1/4$ で求められ、 $0.250\mu s/INST$ である。尚、図2(a)に示すPC0、PC3、PC2、PC3...PC0は、実行順序に従って各PCを並べたものである。

【0032】図2(b)のPC3、PC2、PC3...PC0は、図2(a)に示すPCの一部を示したものである。また、SCHDカウンタ部4からの信号D1「11, 10, 11...00」は、PC3、PC2、PC3...PC0に相当するように「3, 2, 3...0」を2ビットの信号で示したものである。また、図2(d)の0乃至7は、SCHD選択部3からPC選択部5に供給される3ビットの信号D1を意味する。上記構成のマイクロプロセッサにおいては、ユーザは複数のプログラムの処理間隔(時間)を一命令単位でSCHDレジスタファイル部2に設定出来る。

【0033】また、中央処理装置は上記のような使用時間及び使用速度で使用され、レジスタに設定された配列順で繰り返しPCを動作させることにより、各PCはSCHDに設定されたデータにより定まる一定時間おきに実行される。これより、必要に応じてプログラム処理の

6

高速化が出来る。また、各々のプログラムに応じた処理時間の配分が出来るので、システムの効率が図れる。

【0034】尚、SCHDレジスタファイル部2内の選択されるべきSCHDのデータの設定(2ビット×8個)は、プログラム実行中に必要に応じて命令により変えることができる。これにより、プログラムの実行順序を適宜プログラム実行中に切り換えることが出来る。

尚、本発明は上記実施例に限定されず種々の変更が可能である。例えば、上記実施例ではSCHDを4つ設けたが、2つ以上あればよい。また、上記実施例ではPCを4つ設けたが、2つ以上でもよい。

【0035】例えば、SCHD2を選択している際にSCHD設定部1は、バスライン17からのコマンドに従って、SCHD1, 3, 4を自由に書き換える。また、SCHD選択部3はコマンドに従って、SCHDを自由に選択して、例えばSCHD2からSCHD3に切り換える。これにより、プログラムカウンタPCの選択順序がプログラム実行中に切り換えられる。尚、プログラムカウンタPCの選択中の内容自体を書き換えられるようにしてもよい。

【0036】

【発明の効果】上記構成より、CPUが1つ備えられたマイクロプロセッサにおいても、ユーザが複数のプログラムの処理時間を一命令単位で設定出来るので、必要に応じたプログラムを高速で実行出来、プログラムに応じた処理時間の配分が出来ると同時にシステム全体の効率化が図れる。また、プログラムの処理時間が明確化でき、マイクロプロセッサにおけるリアルタイム(並列)処理を容易に設計出来る。

【図面の簡単な説明】

【図1】本発明の実施例に係るマイクロプロセッサのブロック図である。

【図2】図1のマイクロプロセッサのPCの実行例を示す図である。

【符号の説明】

1...SCHD設定部、2...SCHDレジスタファイル部、3...SCHD選択部、4...SCHDカウンタ部、5...PC選択部、6...PC、D1...SCHDカウンタ部のセレクト信号、D2...SCHD選択部の出力信号、10...マイクロプロセッサ、11...I/O、12...ROM、13...RAM、14...ALU、15...PC、16...SCHDブロック、17...バスライン、18...CPU部。

